Extreme Learning Machine design for dealing with unrepresentative features

Nicolás Nieto[a,b], Francisco J. Ibarrola[a], Victoria Peterson[b], Hugo L. Rufiner[a], Ruben Spies[b]

[a] *Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional, sinc(i), UNL-CONICET, FICH, Ciudad Universitaria, CC 217, Ruta Nac. 168, km 472.4, (3000) Santa Fe, Argentina.*

[b] *Instituto de Matemática Aplicada del Litoral, IMAL, UNL-CONICET, Centro Científico Tecnológico CONICET Santa Fe, Colectora Ruta Nac. 168, km 472, Paraje "El Pozo", (3000), Santa Fe, Argentina.*

Email addresses: nnieto@sinc.unl.edu.ar - Argentina (Nicolás Nieto), fibarrola@sinc.unl.edu.ar - Argentina (Francisco J. Ibarrola), vpeterson@santafe-conicet.gov.ar - Argentina (Victoria Peterson), lrufiner@sinc.unl.edu.ar - Argentina (Hugo L. Rufiner), rspies@santafe-conicet.gov.ar - Argentina (Ruben Spies)

Abstract

Extreme Learning Machines (ELMs) have become a popular tool for the classification of electroencephalography (EEG) signals for Brain Computer Interfaces. This is so mainly due to their very high training speed and generalization capabilities. Another important advantage is that they have only one hyperparameter that must be calibrated: the number of hidden nodes. While most traditional approaches dictate that this parameter should be chosen smaller than the number of available training examples, in this article we argue that, in the case of problems in which the data contain unrepresentative features, such as in EEG classification problems, it is beneficial to choose a much larger number of hidden nodes. We characterize this phenomenon, explain why this happens and exhibit several concrete examples to illustrate how ELMs behave. Furthermore, as searching for the optimal number of hidden nodes could be time consuming in enlarged ELMs, we propose a new training scheme, including a novel pruning method. This scheme provides an efficient way of finding the optimal number of nodes, making ELMs more suitable for dealing with real time EEG classification problems. Experimental results using synthetic data and real EEG data show a major improvement in the training time with respect to most traditional and state of the art ELM approaches, without jeopardising classification performance and resulting in more compact networks.

Keywords: Brain Pattern Recognition, Brain Computer Interfaces, Pruning, Unrepresentative Features, Electroencephalography
2010 MSC: 68T05

1. Introduction

Brain-Computer Interfaces (BCIs) have become attractive as they provide alternative ways of communication for people who have lost the capability to interact with their environment (Wolpaw et al., 2002). By means of a BCI, the neural activity of a person is decoded and transformed into commands, which are then used for controlling a device (Nicolas-Alonso & Gomez-Gil, 2012; Holz et al., 2015). In BCI applications, neural activity is classically measured by electroencephalography (EEG), since it is a non-invasive technique, measurement devices can be portable and the EEG signals have high time resolution (Wolpaw et al., 2002; Nicolas-Alonso & Gomez-Gil, 2012). However, those signals present a low spatial resolution, high redundancy and the information is usually encoded only in a small subset of the features. Once the EEG signal is obtained, machine learning techniques are typically used to classify them for their posterior use as inputs for control commands. This classification process presents several challenges, as it needs to yield good performance, its training time must be short and it must be applicable even when only a few training samples are available. In this context, Extreme Learning Machines (ELMs) had become a popular tool for EEG signal classification in BCIs applications (Liang et al., 2006; Duan et al., 2016; Zhang et al., 2018; Kong et al., 2018; Tan et al., 2016; Jin et al., 2020).

The use of random weights in neural networks was first proposed in (Schmidt et al., 1992). The idea was then reintroduced under the concept of Extreme Learning Machines by (Huang et al., 2004, 2006, 2011), and it has been

extensively used since then, not only for BCI applications, but also in the whole Neuroscience field (Murugavel & Ramakrishnan, 2016; Song & Zhang, 2013; Yuan et al., 2011; Zhao et al., 2018; Shi & Lu, 2013). Moreover, ELMs have been recently incorporated into Deep Learning frameworks for EEG classification, resulting in deep architectures with short training times (Ding et al., 2015, 2017). The main reasons for their wide use are that the training process is very fast and the networks yield good generalization capabilities. Another quite appealing aspect of ELMs is that, unlike most machine learning techniques, they have only one hyperparameter that must be tuned up: the number of hidden nodes. In (Huang & Babri, 1998) it has been proved that given a training dataset consisting of N samples, the model can learn them exactly, with probability 1 (w.r.t. the random parameter initialization), using N hidden nodes. Nevertheless, perfect classification over a training set often entails a loss of generalization capabilities. In fact, for the case of N training samples, according to (Huang & Babri, 1998), N is an upper bound for the number of nodes in the hidden layer. However, for the particular case of problems in which the data are contaminated by unrepresentative features, like EEG signals, choosing the number of hidden nodes $M \leq N$ turns out to be suboptimal, since $M \gg N$ will almost surly results in better classification performances. In Sections 2 and 3, we use synthetic data to show this phenomenon and describe the reason for its occurrence.

Enlarged architecture ($M \gg N$) are most often unfeasible in real time applications with traditional ELM training schemes. This is so because training time rapidly increases with the number of nodes, and hence hyperparameter search can become time consuming, thus antagonizing one of the most important ELMs advantages. To cope with the problem of finding the optimal number of hidden nodes, different pruning methods have been developed (Rong et al., 2008; Miche et al., 2010; Luo et al., 2013; Tavares et al., 2014; Alencar et al., 2016). In (Rong et al., 2008) the Information Gain and Chi-Square are computed and used to determine the relation between the hidden nodes and the target labels. In (Miche et al., 2010) the authors propose a method named Optimally Pruned ELM (OP-ELM). This algorithm ranks the nodes using multiresponse sparse regression (MRSR) (Similä & Tikka, 2005), and then prunes the hidden layer using the leave-one-out (LOO) validation error. In (Luo et al., 2013) the authors propose to find a sparse representation of the output weights, effectively pruning the hidden layer. However, this approach uses Bayesian inference instead of the generalized inverse to obtain the output weights, and therefore it can hardly be considered an ELM. In (Tavares et al., 2014) the authors propose to pre-prune linearly dependent nodes, disregarding any other information such as discriminative power. More recently, a pruning method based on genetic algorithms (GA) was introduced in (Alencar et al., 2016). Here, a fitness function simultaneously minimizes the LOO validation error and the number of hidden nodes. However, as this method has to train a different ELM for each individual in every generation, it ends up being computationally expensive.

All previously described pruning methods need to retrain the ELMs for searching their internal hyperparameters and/or have to retrain the whole network after the optimal number of hidden nodes is reached. Furthermore, the methods focus their attention on improving classification performance, with complete disregard for the computational burden of tuning the hyperparameters, which can be very heavy in enlarged ELMs. In order to cope with this drawback, in this article we introduce a novel post-training pruning method. Our proposal changes the traditional way of searching for the number of hidden nodes in enlarged ELMs, eliminating the need for the computational

expensive retraining and without adding extra hyperparameters. The method is described in Section 4. In Section 5, we compare our scheme with the traditional ELM approach with two different EEG databases, and show its potential in real-life problems. Finally, a detailed comparison between our proposed algorithm and the OP-ELM is also presented.

2. Extreme Learning Machines

For simplicity, we shall consider an ELM within the context of a binary classification problem. We point out, however, that all results presented in this work remain valid for multi-class problems.

Given an arbitrary vector $x \in \mathbb{R}^D$, an ELM classification output is given by

$$z = \beta^T g(Wx + b), \tag{1}$$

where $W \in \mathbb{R}^{M\times D}$ is the matrix associated to the hidden layer, $g : \mathbb{R} \to \mathbb{R}$ is an activation function, $b \in \mathbb{R}^M$ is the bias vector, and $\beta \in \mathbb{R}^M$ is the weight vector connecting the hidden layer to the output. Here and in the sequel, the action of g on a vector or on a matrix is meant to be its components-wise evaluation.

The training process of an ELM consists of two main steps. First, the entries of W and b are randomly generated as independent realizations of an absolutely continuous random variable (usually with uniform distribution in $[-1, 1]$). The second step consists of finding an appropriate vector β. This can be done as described below.

Let us consider a dataset consisting of N training samples $x_n \in \mathbb{R}^D$, $n = 1, \ldots, N$, stacked as the columns of a matrix $X \in \mathbb{R}^{D\times N}$. Let $y \in \{-1, 1\}^N$ be the desired output vector, where y_n, gives account for the class of x_n, $\forall n$.

Let us define

$$H \doteq [g(WX + b\,1_{(1,N)})]^T, \tag{2}$$

where $1_{(1,N)}$ is an N-dimensional row vector with all its elements equal to 1. Then, training the ELM weight vector β amounts to "solving" the linear system

$$H\beta = y. \tag{3}$$

Note that $H \in \mathbb{R}^{N\times M}$, and therefore if $M < N$, (*i.e.* if the number of nodes is less than the number of training examples) the linear system (3) is under-determined. In this case, an appropriate way to deal with equation (3), is to resort to least squares solutions. We then define the vector $\hat{\beta}$ as the best approximate solution (i.e, the minimal-norm least-squares solution) of (3) (Heinz et al., 1996). That is

$$\hat{\beta} \doteq H^\dagger y, \tag{4}$$

where $H^\dagger$ is the Moore-Penrose generalized inverse of H.

When $M < N$, the ELM's generalization capability is associated with the fact that least squares solutions assign larger weights to the columns of H which are most relevant for classification purposes. However, when $M = N$,

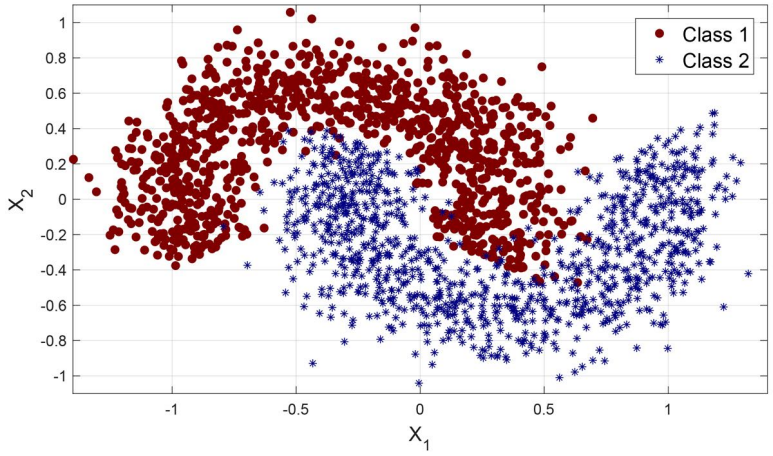

Figure 1: Synthetic data distribution, differentiated by class. 500 data points for each class are plotted

system (3) has a unique solution (with probability one), as shown in (Huang & Babri, 1998). Hence, in this case, the solution $\hat{\beta}$ is forced to take all columns of H into account, even those which are irrelevant for classification. This constitutes a classic case of overfitting.

Although it is theoretically true that $M \geq N$ implies that the matrix H has N independent column vectors (with probability one), this does not necessarily imply that the feature space be well represented. To illustrate this, let us consider the following example. Suppose that there are two columns of H, say h and $h' \in \mathbb{R}^N$, such that $h_1 = h'_1 + \epsilon$ (for a small $\epsilon \in \mathbb{R}$) and $h_n = h'_n, \forall n \geq 2$ (here h_i, h'_i denotes the i^{th} components of the vectors h and h', respectively). Although these two vectors are strictly different and linearly independent, for all practical purposes they clearly encode the same feature information. Formally, while $M = N$ ensures that H be invertible, it can still present very small singular values (and therefore a high condition number), which is a reflection of a poor representation of the feature space. As shown in (Horn & Johnson, 1990), as the smaller non-zero singular values increase, in a context of normalized data, the feature space becomes better represented.

3. Changing the network size

In light of the above discussion, we argue that the proposed upper bound $M \leq N$ is suboptimal in the case of problems where the data have unrepresentative features. Thus, in this kind of problems, such as classification of EEG signals, M should be chosen and calibrated differently. We perform an experiment using artificially generated data in order to characterize the type of problems where choosing $M \gg N$ can be highly beneficial.

Let us consider a binary classification problem in which the data points corresponding to the two classes are distributed as depicted in Figure 1. It is clear that the displayed coordinates x_1, x_2 are enough for classification. In an ideal situation, any additional coordinate added to the data points taking random values independently of the class, should not be taken into account by any classifier. In a real problem, the data points might be highly contaminated with this kind of unrepresentative "junk features" (JF) which are, *a-priori*, indistinguishable from the representative ones. Hence a question that immediately arises is: should the same kind of ELM architecture be

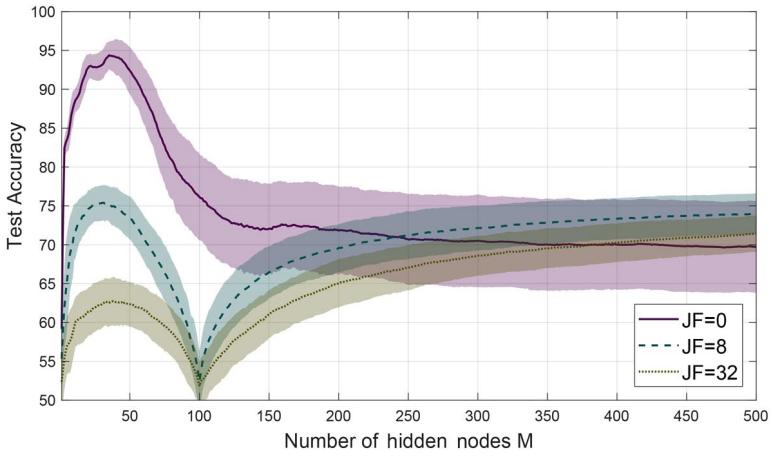

Figure 2: Mean test accuracy for 300 realizations (30 random initializations of parameters W and b and 10 cross validations of the data) in synthetic data adding Junk Features (JF) generated from a uniform distribution in the interval [0,1]. 100 data points are used for training. Shading illustrates ± 1 standard deviation.

used for this kind of problems?

Let us take a look at Figure 2, where for several choices of the number of neurons M, the average test accuracy of an ELM is plotted. The purple solid line corresponds to the test accuracy obtained using the data as displayed in Figure 1, while the other two dashed lines correspond to those obtained using the data points contaminated with different numbers of junk features. Those features were generated as random realizations of a uniform distribution in the interval (0, 1). One can immediately see that $M < N$ is an optimal choice for the two-coordinates case. However, when the data contain unrepresentative features, the choice of $M \gg N$ could be more convenient. As the number of unrepresentative features increases, this convenience becomes more notorious. Different distributions of the junk features have different impacts in the described phenomenon. This can be clearly observed in Figures SI-1, SI-2 and SI-3 of the Supplementary Information Section, where analogous results are shown for different distributions of the unrepresentative features.

One might wonder, if this is a scenario one might often expect in practical problems. As pointed out before, when working with EEG signals, the extracted features will, in most cases, not be representative of the problem at hand, resulting in many junk features. We show that this happens to be the case for an experiment using the datasets introduced in (DaSalla et al., 2009) and (Ledesma-Ramirez et al., 2010), which contain EEG signals. Both datasets are explained in detail in Section 4.1).

The experiments were performed using 70% of the data for training and 30% for testing, with 50 random initializations of the parameters W and b and 20 cross validations. Figure 3 illustrates the train and test accuracy of the ELM as a function of the parameter M for the dataset introduced in (DaSalla et al., 2009). As it can be seen, at first the accuracy grows with M, until reaching a local maximum, after which it starts to decay up to the global minimum, reached at $M = N$. However, the test accuracy starts to increase again as M further increases. Similar results are shown in Figure SI-4 of the Supplementary Information Section, obtained for the dataset introduced in (Ledesma-Ramirez et al., 2010). These results are consistent with the analyzed synthetic data and also with

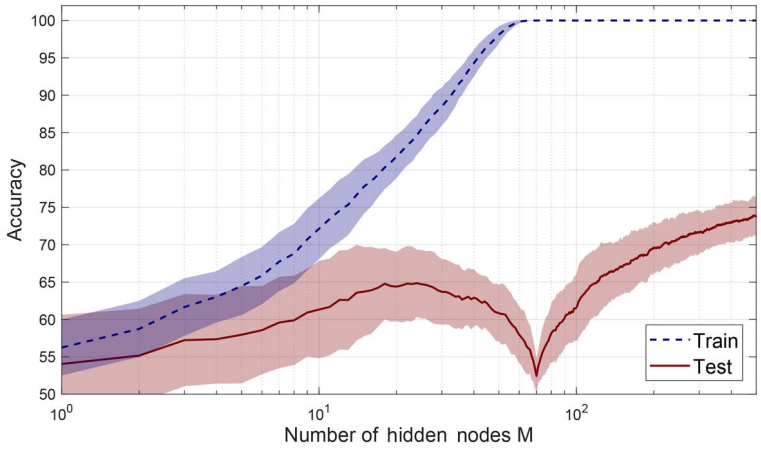

Figure 3: Mean accuracy for 1000 realizations (50 randoms initializations and 20 cross validations) obtained for training and testing data, as a function of the hidden layer size, M. 70 training examples for DaSalla dataset were used. Shading illustrates $\pm$ 1 standard deviation of the results.

the findings in (Belkin et al., 2019), where a similar behaviour is reported in the context of traditional multi-layer perceptrons (MLP). Although this phenomenon is well described and known for MLP, to our knowledge, it has never been previously studied in the context of ELMs.

The overfitting observed in Figures 2 and 3 when $M \approx N$ can be explained by the fact that when training under this condition, we are forcing the network to take into account all the nodes, even those that are irrelevant for classification proposes. To corroborate that this is in fact the reason, we have performed an experiment consisting of adding a disconnected "fake" neuron to the ELM (before training using the DaSalla dataset). That is, a column of random elements having no correlation with the classes was stacked to the right of the matrix H. Figure 4 depicts the absolute value of the weight that the model assigns to this disconnected neuron as a function of the number of neurons M. As seen, this weight remains small until M approaches N, which supports our previous hypothesis. It is timely to observe, however, that the weight assigned to the fake neuron starts to decay again after this point. This means that for $M > N$, the ELM becomes capable of neglecting the value of the disconnected neuron, since it is irrelevant for classification purposes.

4. Relevance-based pruning

As we have shown, in the cases of unrepresentative features, an ELM can benefit from choosing $M \gg N$. Yet this has the downside of increasing the network size. In order to find a compromise between ELM size and classification performance, one could use a traditional validation method and retrain the network increasing the number of neurons M until the change in performance is small enough to be neglected. Nevertheless, the computational burden associated with solving (4) increases with M, and the sensitivity of the method with respect to the random values might require a few initializations, making this idea unfeasible in practice. Hence, in order to make ELMs of practical use for real data in real applications, like EEG signals classification in BCI devices, we propose a novel

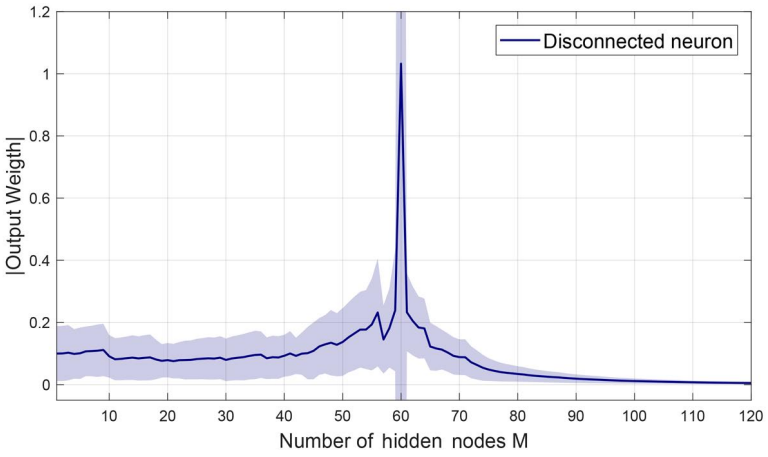

Figure 4: Mean absolute value for 1000 realizations (50 random initializations and 20 cross validations) of the weight β assigned to a "fake" neuron of the hidden layer. Shading illustrates $\pm$ 1 standard deviation of the results.

pruning method which allows for an efficient search of the parameter M, by appropriately reducing the network size after just a single computation of (4) for $M \gg N$.

As observed in the description of Figure 4, the weight of a neuron is proportional to the relevance of the corresponding node. Hence, it is reasonable to discard the neurons whose associated weights β are small enough. Given that the process of discarding a neuron and testing the performance of the resulting ELM is computationally inexpensive, the proposed pruning method begins using a large initial number M^* of hidden nodes, and then discard one (or a few) at a time, just until before the performance exhibits a significant drop. We shall refer to the resulting method as Relevance-Based Pruning (RBP). The steps for performing RBP are shown in Algorithm 1. In Section 4.1 we present different experiments in order to validate the pruning criterion and in Section 5, comparisons between this pruning method, the standard $M < N$ setting, the traditional validation froward scheme and the OP-ELM, in terms of classification performance and computational costs, are also shown.

4.1. Pruning criterion validation experiments

4.1.1. Experimental setting

Two EEG datasets were used for the experiments. As previously described, EEG data typically presents high levels of noise, and the relevant classification information is mostly encoded in a particularly small subset of features. The first dataset is the imagined speech dataset, introduced in (DaSalla et al., 2009). It contains EEG signals recorded from three different subjects using 64 electrodes with a sampling frequency of 256 Hz. Here, we used the spatially filtered EEG signals related to imagination of mouth movement involved in the pronunciation of two vowels (/a/ and /u/) and a resting state, in pairwise comparisons (C1 = /a/ vs. rest; C2 = /u/ vs. rest; C3 = /a/ vs. /u/). 100 trials were presented for every comparison, with 512 features each. The second dataset is a P300-based BCI dataset introduced in (Ledesma-Ramirez et al., 2010), consisting of 3780 EEG trials of one second (630 trials with P300 activity), acquired from 25 subjects using 10 channels at 256 Hz. Each trial has 2560 features (256 samples x 10 channels) and each trial label corresponds to the presence or absence of P300 activity.

Algorithm 1 : Relevance-Based Pruning (RBP)

Set $M^* >> N$ and $\delta > 0$.

Initialize the elements of $W \in \mathbb{R}^{M^* \times D}$ and $b \in \mathbb{R}^{M^*}$ randomly as realization of a distribution $\mathcal{U}[-1, 1]$, M^* being the maximum number of hidden nodes and D the data dimension.

$H = [g(WX_{train} + b\,1_{(1,N)})]^T$.

$\hat{\beta} = H^{\dagger} y_{train}$.

$H = [g(WX_{val} + b\,1_{(1,N)})]^T$.

Permute $\hat{\beta}$ so that $|\hat{\beta}_m| \geq |\hat{\beta}_{m+1}|,\ \forall m = 1, \ldots, M^*$.

Perform the same permutation on the columns of H.

Let $H' = H$, $\hat{\beta}' = \hat{\beta}$

while $\text{mean}(H\hat{\beta} - y_{val}) < \text{mean}(H'\hat{\beta}' - y_{val}) + \delta$ **do**

Let $H' = H$, $\hat{\beta}' = \hat{\beta}$

Remove the last element from $\hat{\beta}$.

Remove the last column from H.

end while

Algorithm 2 : Standard ELM forward scheme

Set $M = 1$ and $\delta > 0$.

Set $acc_{val} = 0$ and $acc'_{val} = 0$.

while $M < M^*$ and $acc_{val} > acc'_{val} + \delta$ **do**

Initialize the elements of $W \in \mathbb{R}^{M^* \times D}$ and $b \in \mathbb{R}^{M}$ randomly as realization of a distribution $\mathcal{U}[-1, 1]$.

$H = [g(WX_{train} + b\,1_{(1,N)})]^T$.

$\hat{\beta} = H^{\dagger} y_{train}$

$acc'_{val} = acc_{val}$

Compute acc_{val} using X_{val} and y_{val}.

$M \leftarrow M + 1$.

end while

4.1.2. Classification performance analysis

In order to compare RBP against the standard "forward" ELM approach, we propose the following experimental setting: consider the datasets $X_{train} \in \mathbb{R}^{D\times N}$ and $X_{val} \in \mathbb{R}^{D\times N'}$ and a maximum ELM size M^*. For the traditional method, the steps followed are shown in Algorithm 2. It is worth mentioning that in these experiments, we aim to validate the pruning criterion rather than to find the optimal parameter M of the network. Therefore, the data were only split in train and test sets. Also, the Algorithms 2 and 1 ignore the stop parameter δ.

Using 50 random initializations for W and b, for every value of $M \in \{1, \ldots, M^* = 1000\}$, we run the proposed experiment for a randomly chosen subject of the DaSalla dataset, over 20 cross validations with a 70/30 train/test scheme. The resulting average validation accuracy for both the standard forward method (Algorithm 2) and the RBP method are depicted in Figure 5 (top). The results obtained using a random pruning scheme (RND) are used to compare and validate the proposed pruning criterion.

An analogous test was made using a (randomly chosen) subject of the P300 database. Results are shown in Figure 5 (bottom). Since in this case the dataset is much larger, instead of taking unitary increments on the values of M, a logarithmic grid was used. Also, as the dataset is unbalanced, the area under the ROC curve (AUC) was used to evaluate the classification performance. The results correspond to 5 cross validations over 20 random initializations, with a choice of $M^* = 40000$.

As seen in Figure 5, RBP performs at least as good as the forward scheme. This means that choosing a large value of M^* followed by a reduction of the network size by means of RBP (until just before a significant decay in validation performance is observed) will yield a result as good as using the forward scheme, and that can be done in a very short time, as the pruning procedure is computationally inexpensive. Thus, we can choose M^* large enough (e.g. as determined by the computational cost we are willing to pay), and then run RBP to reduce the network size and quickly find the optimal parameter M without losing classification performance, so ending up with a compact network, which is specially important in real time applications. Finally, the random pruning method validates our hypothesis that the less relevant nodes can be pruned with little or no impact in the classification performance.

5. Analysis of RBP global performance on real data

In order to validate our proposal for finding the optimal number of hidden nodes M in real applications, we made comparisons with four different schemes: STD, FWD, RBP, OP-ELM, which are described below :

STD: (traditional approach) best result obtained using $M < N$.

FWD: start with $k = 0$ and $M_k = 1$. Increase M_k over a logarithmic grid until the validation performance levels up or the maximum value M^* is reached (see Algorithm 2).

RBP: start with $k = 0$ and $M_k = M^*$ and prune M_k over a logarithmic grid until just before the first time classification performance shows a significant (prescribed) reduction. The stopping criterion, grid and the value of M^* are set equal to those in FWD.

OP-ELM: the publicly available implementation of OP-ELM was used with $M = M^*$ and $M = N$.

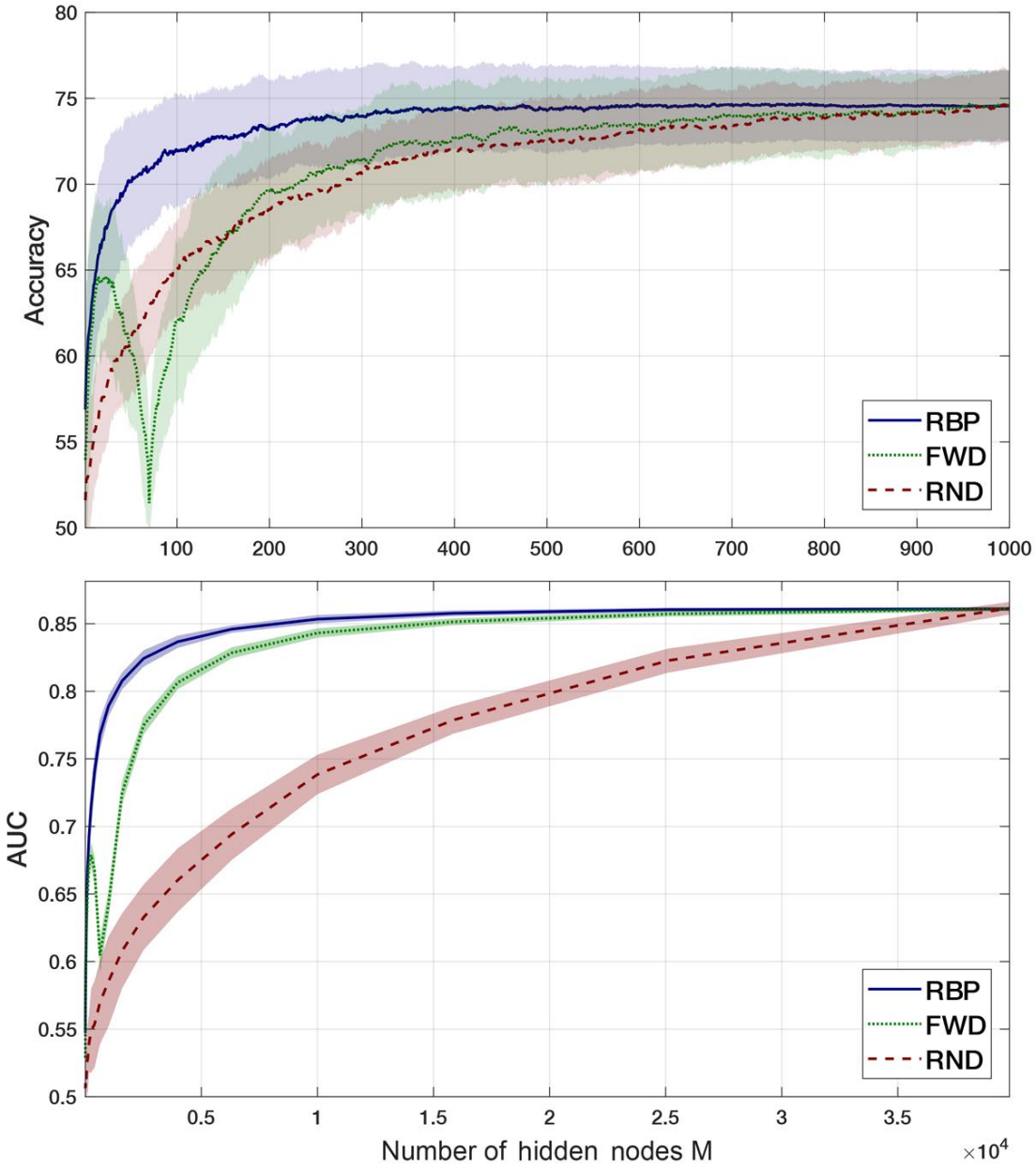

Figure 5: Average classification performance obtained with the traditional forward scheme (FWD), Random Pruning (RND) and Relevance-Based Pruning (RBP) methods, for one subject of the DaSalla dataset (top) and one subject of the P300 database (bottom). 70 and 2646 training examples were used for DaSalla and P300 experiment respectably. Shading accounts for ± one standard deviation.

Method	Accuracy (%)	Time [ms]	M
STD	62.2 ± 7.3	76 ± 48	24 ± 18
FWD	66.9 ± 5.5	195 ± 21	228 ± 76
OP-ELM $M^* = 90$	63.5 ± 14.8	33.9 ± 9	27 ± 14
OP-ELM $M^* = 1000$	63.1 ± 14.3	23471 ± 54	77 ± 10
RBP	68.0 ± 3.8	43 ± 3	209 ± 151

Table 1: Overall performance results yielded by each tested method for the DaSalla database.

Method	AUC	Time [s]	$M[\times 10^3]$
STD	0.718 ± 0.072	3 ± 1	0.5 ± 0.2
FWD	0.816 ± 0.096	572 ± 118	28.2 ± 10.6
RBP	0.816 ± 0.087	194 ± 1	14.5 ± 6.5

Table 2: Overall performance results yielded by each tested method for the P300-based BCI dataset. Figure 6 also illustrates the performance of the five first subjects

For our first experiment the full DaSalla dataset (three subjects in three comparisons) was used. For all methods, 5 random initializations over 20 cross validations were run. While for STD, FWD and RBP the data were split in 70%, 20% and 10% for training, validation and testing, respectively, for OP-ELM, 90% training and 10% testing were used, as the validation split is automatically made within the implementation. As the dataset is balanced, accuracy was used to measure classification performances. For FWD and RBP, the parameters were set to $M^* = 1000$ and $\delta = 0.02$. Table 1 shows the obtained results, from which three main conclusions can be derived. First, both methods allowing an ELM with a large number of hidden nodes result in better performances, in terms of accuracy. Secondly, between those two approaches, RBP is observed to require much less computing time with a comparable network size. Finally, OP-ELM is not able to handle this type of problems, as the required time is several orders of magnitude higher and it does not yield better classification performance. As the OP-ELM method is not designed nor suggested for the case $M > N$, nor for the case of data with high redundancy, like EEG data, in the sequel, OP-ELM was neglected in all the analyses. A subject by subject analysis is presented in the Supplementary Information Section in Figures SI-5 and SI-6.

The same experiment was performed, but now using the P300-based BCI dataset. The data was also split into 70%, 20% and 10% for training, validation and testing. For each one of the 25 subjects, five cross validations over 20 random initializations were performed. The parameters were set to $M^* = 40000$ and $\delta = 0.001$. As previously mentioned, the P300-based BCI dataset is unbalanced, so the AUC was used for measuring classification performance. Overall results are shown in Table 2, and those obtained for the first five subjects are illustrated in Figure 6, along with overall performance.

As it can be seen, the AUC values obtained by FWD and RBP, *i.e.* the methods allowing for $M > N$, perform considerably better than those obtained with STD. While FWD and RBP do not account for practical differences in

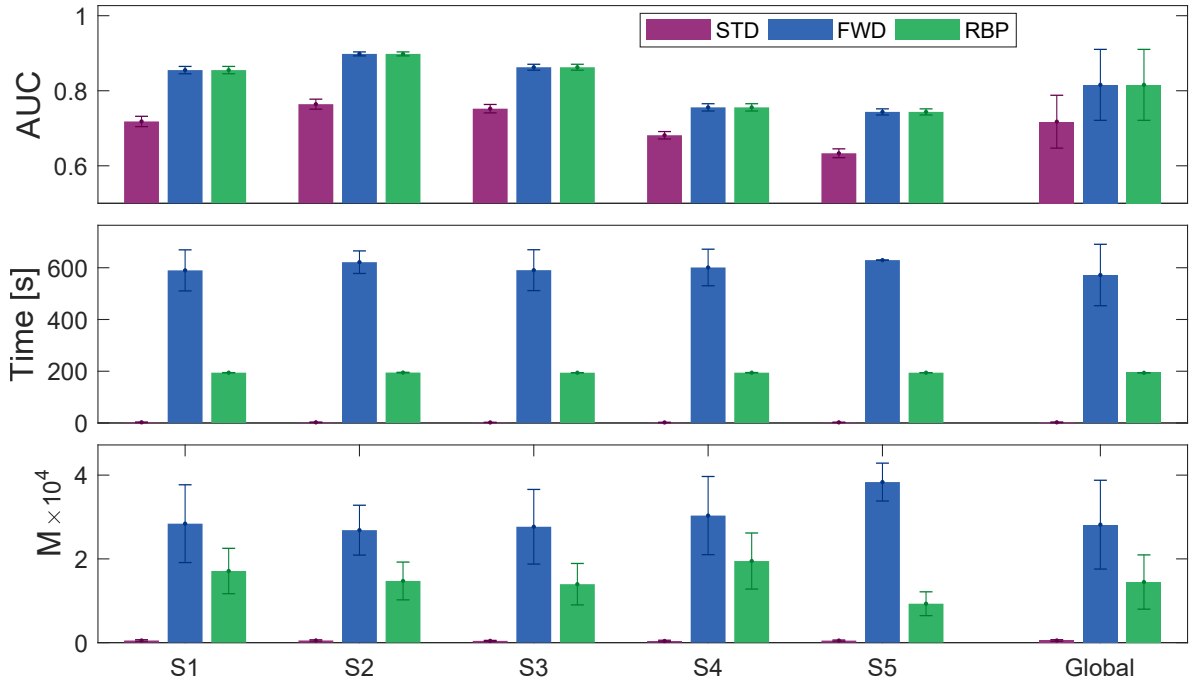

Figure 6: Results obtained with standard (STD), forward (FWD) and Relevance-Based Pruning (RBP) schemes on the P300-based BCI dataset.

AUC, RBP requires much less computing time than the former. Additionally, RBP yields a much smaller network size resulting in a more compact ELM.

From a practical point of view, if the number of hidden nodes is a restriction, then RBP is the best choice because it will yield better classification performance with the same number of neurons. On the other hand, if only classification is relevant, then RBP is more appropriate since it will yield the same performance in much less training time than FWD with a more compact network.

An experiment was also conducted for measuring the impact of the election of M^* in the training time, as the time performance could strongly depend on its choice. Comparison results for the FWR, RBP and OP-ELM methods are presented in Figures SI-7 and SI-8 of the Supplementary Information Section for the DaSalla and the P300 datasets, respectively. These results clearly show that the proposed RBP method is faster than both the traditional FWD and the OP-ELM algorithm, regardless of the M^* initialization. It is also timely to point out that time gain becomes larger for larger values of M^*. Therefore, the advantage of using the RBP method is greater in the cases of datasets containing a high number of N training examples.

6. Conclusions and future work

In this work we have shown that when dealing with classification problems in the context of data with unrepresentative features, such as EEG signals, ELMs benefit from choosing $M \gg N$. A detailed analysis on why standard approaches are suboptimal for this type of data was provided.

Results using real EEG data show that choosing a large number of hidden neurons is beneficial and that our proposed Relevance-Based Pruning method provides a time-efficient way to search for the optimal number of hidden nodes without jeopardizing classification performance. Furthermore, its implementation is very simple and it can result in great benefits in real application.

In regard to the limitations or constraints of the proposed RBP method we must mention that its use will certainly not lead to any advantage with respect to the other methods, when the data contain just a few unrepresentative features or are completely free of them. This could pose a problem in cases where we have absolutely no information about the existence of junk features in the data.

In the future, we shall tackle the problem of choosing an appropriate maximum ELM size (M^*) depending on the problem. Also, we intend to incorporate this method in the context of regularized ELMs and explore its potential as a feature selection tool.

Conflict of interest

The authors declare that they have no conflict of interest.

Information Sharing Statement

The P300 based BCI dataset is publicly available at `https://akimpech.izt.uam.mx/p300db/`. The DaSalla imagined speech dataset was originally available at `http://www.brainliner.jp/data/brainliner-admin/Speech_Imagery_Dataset`. The synthetic data, along with a Python and MatLab implementation of the Relevance-Based Pruned method are also publicly available to encourage reproducible research and can be accessed at `https://github.com/N-Nieto/Relevance_Base_Pruning`. The OP-ELM implementation was downloaded from `https://research.cs.aalto.fi/aml/software.shtml`.

Acknowledgments

This research was funded in part by Consejo Nacional de Investigaciones Científicas y Técnicas, CONICET, Argentina, through PIP 2014-2016 No. 11220130100216-CO, the Agencia Nacional de Promoción Científica y Tecnológica through PICT-2017-4596 and by Universidad Nacional del Litoral, UNL, through CAI+D-UNL 2016 PIC No. 50420150100036LI.

Bibliography

Alencar, A. S., Neto, A. R. R., & Gomes, J. P. P. (2016). A new pruning method for extreme learning machines via genetic algorithms. *Applied Soft Computing*, *44*, 101–107.

Belkin, M., Hsu, D., Ma, S., & Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, *116*, 15849–15854.

DaSalla, C. S., Kambara, H., Sato, M., & Koike, Y. (2009). Single-trial classification of vowel speech imagery using common spatial patterns. *Neural Networks*, *22*, 1334–1339.

Ding, S., Guo, L., & Hou, Y. (2017). Extreme learning machine with kernel model based on deep learning. *Neural Computing and Applications*, *28*, 1975–1984.

Ding, S., Zhang, N., Xu, X., Guo, L., & Zhang, J. (2015). Deep extreme learning machine and its application in eeg classification. *Mathematical Problems in Engineering*, *2015*.

Duan, L., Bao, M., Miao, J., Xu, Y., & Chen, J. (2016). Classification based on multilayer extreme learning machine for motor imagery task from EEG signals. *Procedia Computer Science*, *88*, 176–184.

Heinz, W., Engl, M. H., & Neubauer., A. (1996). Regularization of inverse problems. *Mathematics and its Applications (Dordrecht)*, *375*.

Holz, E. M., Botrel, L., Kaufmann, T., & Kübler, A. (2015). Long-term independent brain-computer interface home use improves quality of life of a patient in the locked-in state: a case study. *Archives of Physical Medicine and Rehabilitation*, *96*, S16–S26.

Horn, R. A., & Johnson, C. R. (1990). *Topics in matrix analysis*. Cambridge University Press.

Huang, G.-B., & Babri, H. A. (1998). Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE Transactions on Neural Networks*, *9*, 224–229.

Huang, G.-B., Wang, D. H., & Lan, Y. (2011). Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, *2*, 107–122.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, *70*, 489–501.

Huang, G.-B., Zhu, Q.-Y., Siew, C.-K. et al. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. *Neural Networks*, *2*, 985–990.

Jin, Z., Zhou, G., Gao, D., & Zhang, Y. (2020). EEG classification using sparse Bayesian extreme learning machine for brain–computer interface. *Neural Computing and Applications*, *32*, 6601–6609.

Kong, W., Guo, S., Long, Y., Peng, Y., Zeng, H., Zhang, X., & Zhang, J. (2018). Weighted extreme learning machine for P300 detection with application to brain computer interface. *Journal of Ambient Intelligence and Humanized Computing*, (pp. 1–11).

Ledesma-Ramirez, C., Bojorges-Valdez, E., Yáñez-Suarez, O., Saavedra, C., Bougrain, L., & Gentiletti, G. G. (2010). An open-access P300 speller database. In *Fourth International Brain-Computer Interface Meeting*. Asilomar, California, USA.

Liang, N.-Y., Saratchandran, P., Huang, G.-B., & Sundararajan, N. (2006). Classification of mental tasks from EEG signals using extreme learning machine. *International Journal of Neural Systems*, *16*, 29–38.

Luo, J., Vong, C.-M., & Wong, P.-K. (2013). Sparse bayesian extreme learning machine for multi-classification. *IEEE Transactions on Neural Networks and Learning Systems*, *25*, 836–843.

Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., & Lendasse, A. (2010). OP-ELM: optimally pruned extreme learning machine. *IEEE Transactions on Neural Networks*, *21*, 158–162.

Murugavel, A. M., & Ramakrishnan, S. (2016). Hierarchical multi-class SVM with ELM kernel for epileptic EEG signal classification. *Medical & Biological Engineering & Computing*, *54*, 149–161.

Nicolas-Alonso, L. F., & Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors*, *12*, 1211–1279.

Rong, H.-J., Ong, Y.-S., Tan, A.-H., & Zhu, Z. (2008). A fast pruned-extreme learning machine for classification problem. *Neurocomputing*, *72*, 359–366.

Schmidt, W. F., Kraaijveld, M. A., Duin, R. P. et al. (1992). Feed forward neural networks with random weights. In *International Conference on Pattern Recognition* (pp. 1–1). IEEE Computer Society Press.

Shi, L.-C., & Lu, B.-L. (2013). EEG-based vigilance estimation using extreme learning machines. *Neurocomputing*, *102*, 135–143.

Similä, T., & Tikka, J. (2005). Multiresponse sparse regression with application to multidimensional scaling. In *International Conference on Artificial Neural Networks* (pp. 97–102). Springer.

Song, Y., & Zhang, J. (2013). Automatic recognition of epileptic EEG patterns via extreme learning machine and multiresolution feature extraction. *Expert Systems with Applications*, *40*, 5477–5489.

Tan, P., Sa, W., & Yu, L. (2016). Applying extreme learning machine to classification of EEG BCI. In *2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)* (pp. 228–232). doi:10.1109/CYBER.2016.7574827.

Tavares, L. D., Saldanha, R. R., Vieira, D. A., & Lisboa, A. C. (2014). A comparative study of extreme learning machine pruning based on detection of linear independence. In *2014 IEEE 26th International Conference on Tools with Artificial Intelligence* (pp. 63–69). IEEE.

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, *113*, 767–791.

Yuan, Q., Zhou, W., Li, S., & Cai, D. (2011). Epileptic EEG classification based on extreme learning machine and nonlinear features. *Epilepsy Research*, *96*, 29–38.

Zhang, Y., Wang, Y., Zhou, G., Jin, J., Wang, B., Wang, X., & Cichocki, A. (2018). Multi-kernel extreme learning machine for EEG classification in brain-computer interfaces. *Expert Systems with Applications*, *96*, 302–310.

Zhao, H., Guo, X., Wang, M., Li, T., Pang, C., & Georgakopoulos, D. (2018). Analyze EEG signals with extreme learning machine based on pmis feature selection. *International Journal of Machine Learning and Cybernetics*, *9*, 243–249.